2013年12月に仕様を公開した μITRONの次世代版RTOS 「μT-Kernel2.0」のご紹介

2014年2月7日

ユーシーテクノロジ株式会社

ユーシーテクノロジ株式会社のご紹介

- ユーシーテクノロジ株式会社
  - 英語表記：Ubiquitous Computing Technology Corporation
- 事業内容：
  - 最先端のユビキタス・コンピューティング技術を適用したIoT（Internet of Things）やM2M分野へのソリューションを提供
  - 組込みシステムからシステム構築、コンテンツ制作まで様々なテクノロジをご提供
- 所在地
  - 東京都品川区西五反田2-20-1 第28興和ビル
- 設立：2004年9月
- ホームページ
  - http://www.uctec.com

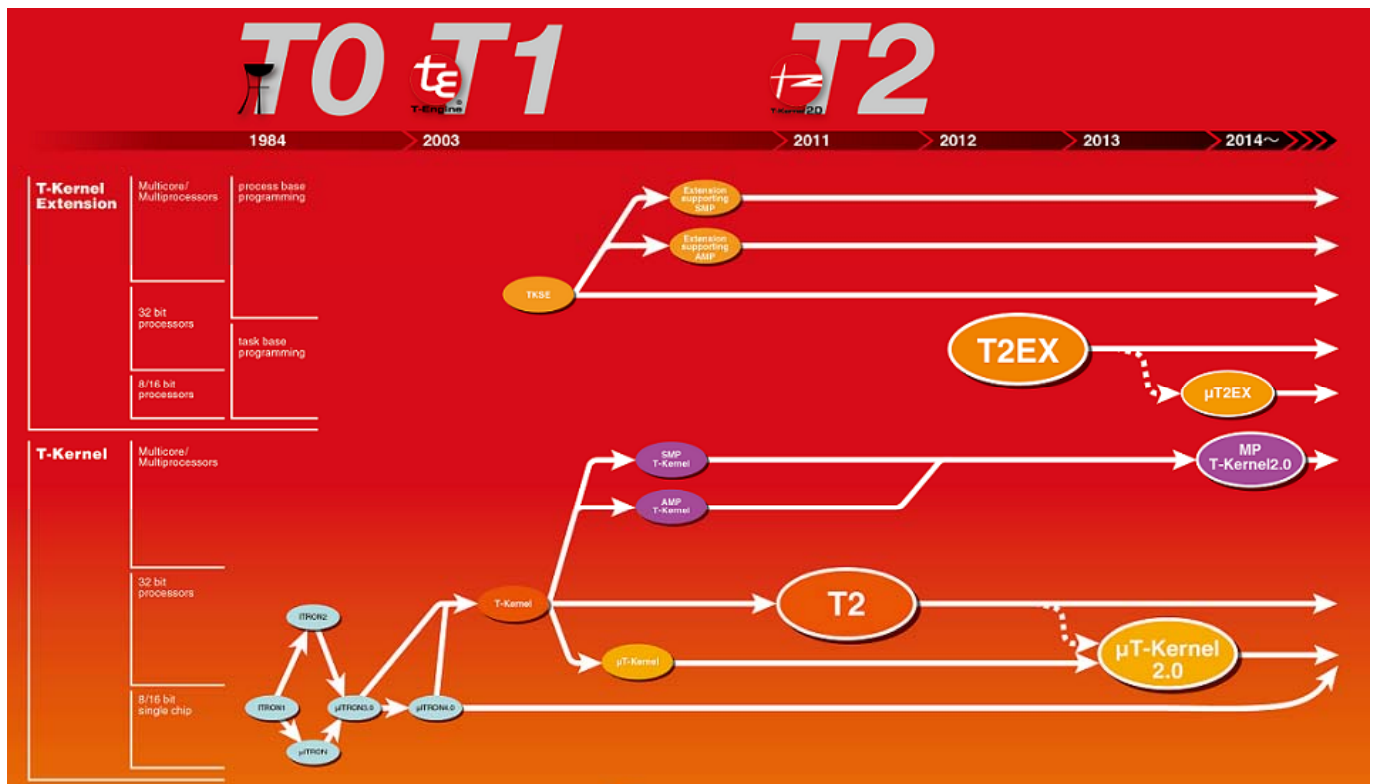
最近のT-Kernelの動向

T-Engine Forumとは?

- ユビキタス・コンピューティングの実現を目指して、ユビキタスIDアーキテクチャとリアルタイム組込みシステムの開発効率向上のための標準化を進める国際的な標準化団体(設立:2002/6 トロン協会は2010年に吸収)
  - T-Kernelの仕様策定とオープンソースコードの提供、およびミドルウェアと開発環境の普及
  - ITRON仕様の策定と保守
  - モノや場所を特定する共通番号であるucodeとユビキタスIDアーキテクチャの仕様策定
  - ucodeの発行/管理を行うユビキタスIDセンターの運営
  - ユビキタス・コンピューティング環境の実現に向けた標準化活動、および政府機関や国際機関との調整
- 約250社の世界の企業・団体が参加
  - ◆会長:坂村健(東京大学教授)

UC Technology

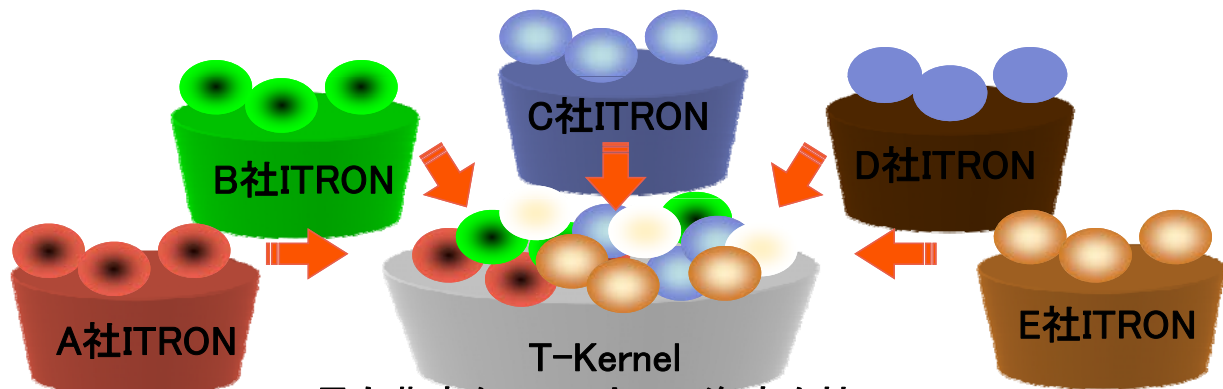
T-Kernel & ITRON Road Map

T-Kernelの特徴は、強い標準化

- ■ハードウェア仕様を規定
  - ◆標準T-Engine、μT-Engine
- ■リファレンスコードを全世界に向けて無償で公開
  - ◆Single One Source

大規模システムも前提としたミドルウェアの流通促進

- ◆ITRONは弱い標準化がウリ
  - ●ハードは規定しない。仕様のみ公開
  - ●1987年当時はPCでさえ16ビットCPUが主流
    - PC-9801VX21, FM-R, PC-286, X68000, IBM PS/2
    - MS-Windows 2.0
  - ●仕様のみ提供していることから実装にバラつきがある

ミドルウェア流通

- ITRON上で、ばらばらに開発されていた、大量のミドルウェア群を共通化して流通
  - CPUの高速化によりOS自体の性能向上より開発効率を重視

最も豊富なソフトウェア資産を持つ
オープンリアルタイムプラットフォームへ！

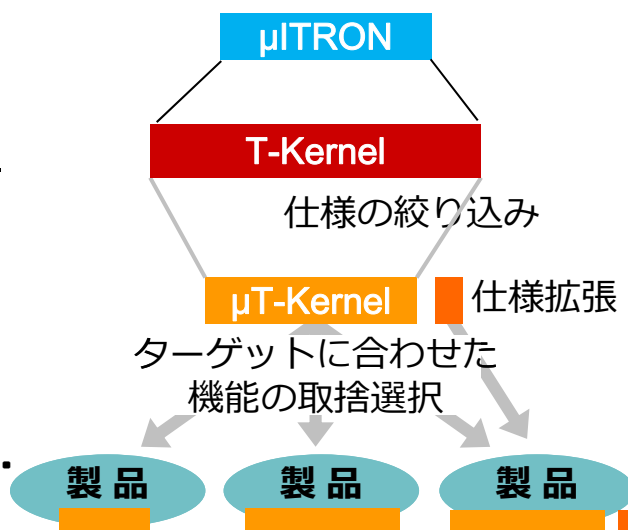
- ソフトウェアパーツの再利用技法を活用した、ソフトウェアの生産性向上

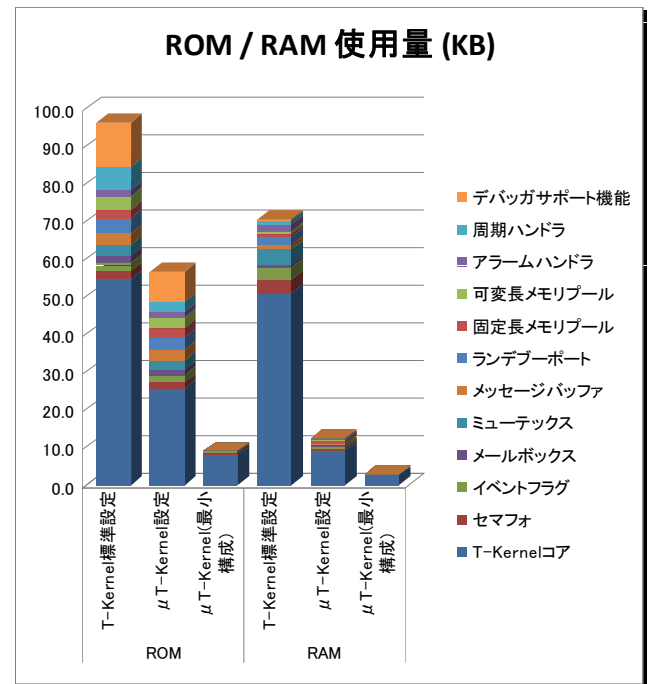
T-Kernelのポイント

- T-Engineフォーラムからオープンソースとして最新仕様RTOSとソースコードを世界に向けて公開
- 会員企業で仕様の改良、メンテナンスを実施
  - 組込みシステムで大きな実績を持つμITRONの技術を継承
  - ソフトウェアの再利用性、流通性を最大に狙う
- 組込み機器に適用しやすいライセンス(T-Licence)で配布
  - 改変したソースコードを公開するのも、あるいは、公開しないのも自由
  - 知的財産権に関して、T-Engineフォーラムの会員各社で内容を確認
- 会員による製品化を行うためにTestSuiteを配布

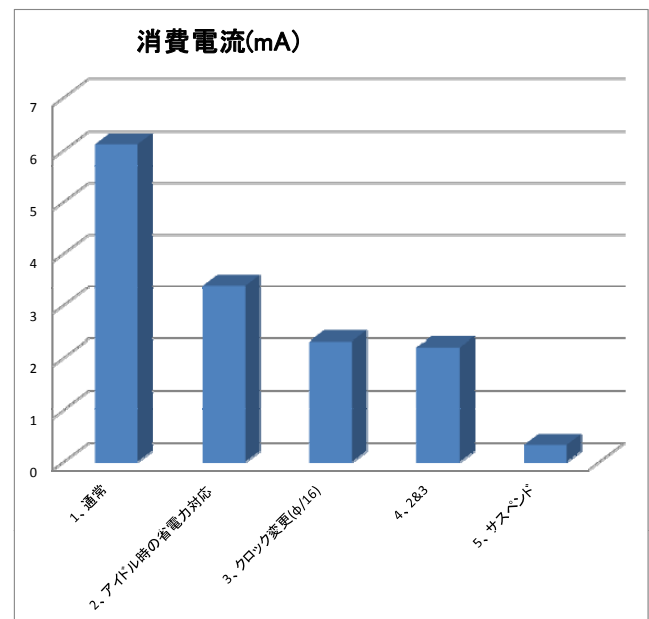
μT-Kernelとは

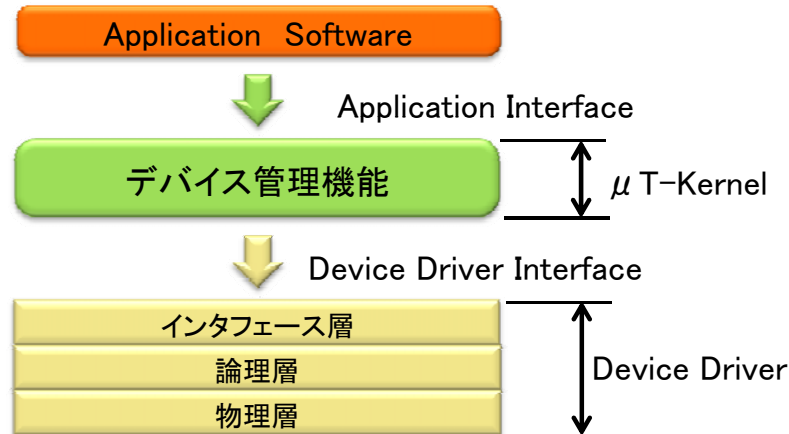
μT-Kernelとは?

- ITRONの後継として小規模システムへの最適化とT-Kernel機能の継承を絶妙なバランスで実現
  - 不要機能削除
    - 必要な機能に絞ったシンプル・カーネル)
  - オーバヘッドになる機能削除
    - 小規模マイコン・アーキテクチャを想定した仕様
  - 資源を有効に使う機能追加

μT-Kernel小規模システム向け対応

- 省資源化に向けた取組み
  - 使用頻度の少ない機能の削除
    - タスク例外機能など
  - 使用しないSVCは取り外す機構
  - メモリ領域の引渡し機能
    - 動的確保のオーバヘッドを削減
  - SVC呼出し方法の変更
    - 関数形式も可能とした
  - ソフトウェア割込みよりも高速処理が可能
- 小さいフットプリントを実現

μT-Kernel省電力機能対応

- 経緯
  - 標準化が難しい
  - 省電力化の情報が少ない
- 省電力機能の実装
  - 4パターンを実装し
    - アイドル時の省電力
    - CPUクロック周波数の低減
    - サスペンド機能の利用
  - 実装によるによる結果
    - 消費電力を1/2～1/10程度に削減可能

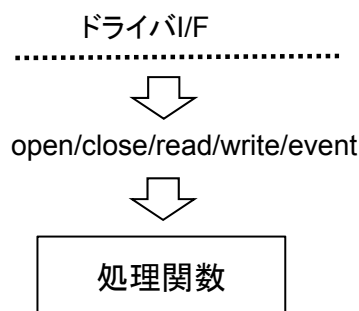
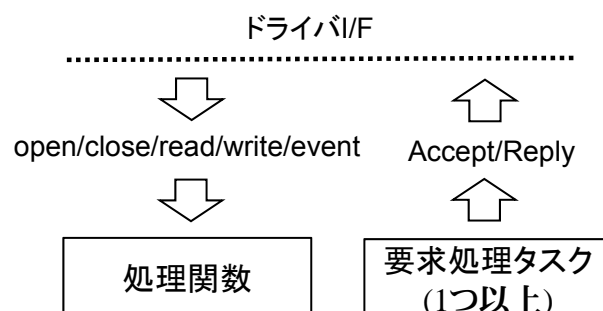
- 「省電力機能実装ガイドライン」
  - T-Engineフォーラム会員に公開

μT-Kernelのデバイスドライバ制御

■ μT-Kernelのデバイスドライバは、保守性/移植性向上のため下図の3層に分けて実装することを推奨

◆デバイスドライバの各層間のインタフェースは、μT-Kernelでは規定していません。
→ 各デバイスドライバに応じた最適な実装が可能

デバイスドライバI/Fライブラリの種類

■ 単純デバイスドライバI/F(SDI:Simple Device Driver I/F)
- ◆すべての処理を待ちに入ることなく即座に処理できる、ごく単純なデバイスドライバを作成する際に使用する。
- ◆例: RTC、タイマ等のレジスタベースの物

■ 汎用デバイスドライバI/F(GDI:General-purpose Device driver I/F)
- ◆要求順に処理する必要のある、一般的なデバイスを対象としたデバイスドライバを作成する際に使用する。
- ◆例: RS-232C 等

μT-Kernelが提供するAPI（例）

■デバイスドライバを利用するためのAPIとして提供

◆どのデバイスドライバでも、同一のAPIで制御可能

■デバイス管理API

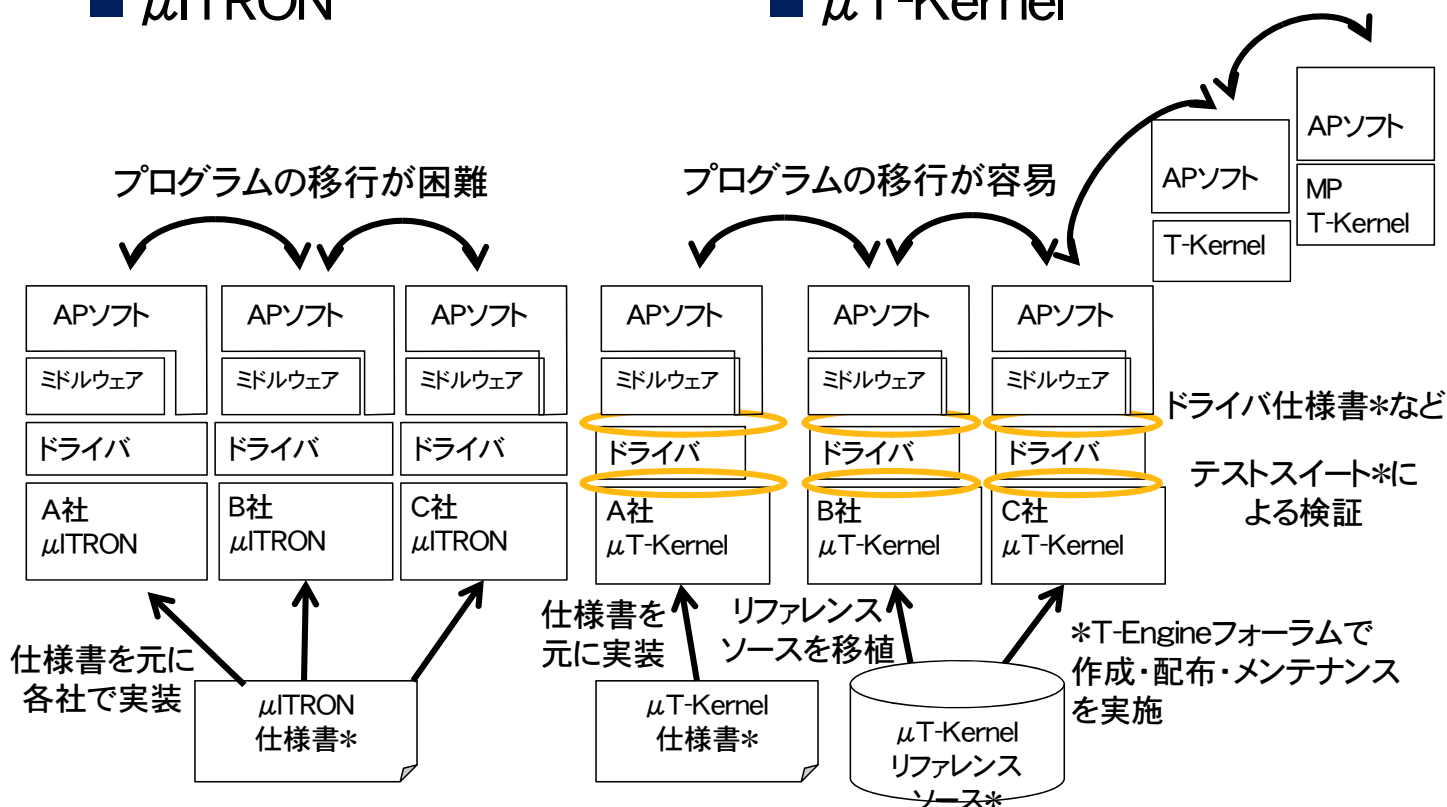
◆tk_opn_dev()　デバイスのオープン
◆tk_cls_dev()　デバイスのクローズ
◆tk_rea_dev()　デバイスの読込み開始
◆tk_srea_dev()　デバイスの同期読込み
◆tk_wri_dev()　デバイスの書込み開始
◆tk_swri_dev()　デバイスの同期書込み
◆tk_wai_dev()　デバイスの要求完了待ち
：　：

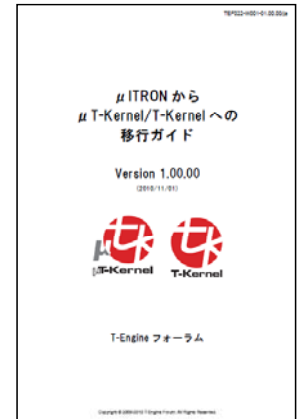
μITRONとμT-Kernel

■ μITRON　■ μT-Kernel

μITRONからμT-Kernelへの移行

μITRON から
μT-Kernel/T-Kernel への
移行ガイド

Version 1.00.00

T-Engine フォーラム

■ 移行のメリット
- ◆アプリケーションプログラムの移植性向上
- ◆流通しているミドルウェア/ドライバ
  - T-KernelとI/Fの統一
  - デバイスドライバ仕様書等
- ◆省電力への対応が容易
  - 会員向けに「μT-Kernel消費電力機能実装ガイドライン」をリリース
- ◆マルチコア環境への移行が容易(MP T-Kernel)

■ 移行方法
- ◆T-Engineフォーラムより「μITRONからμT-Kernel/T-Kernelへの移行ガイド」をリリース
- ◆http://www.t-engine.org/ja/wpcontent/themes/wp.vicuna/pdf/specifications/ja/TEF022-W001-01.00.00_ja.pdf

μITRONからμT-Kernelへの移行(その1)

■ μT-Kernelのシステムコールに置換
- ◆システムコール名の変更
- ◆オブジェクト IDの割り当て方法の変更
- ◆動的APIへの統一
- ◆拡張情報exinf の使い方の統一
- ◆同期・通信オブジェクトなどへの拡張情報exinf の追加
- ◆デバッガサポート機能の追加(dsname)
- ◆類似したシステムコールの統一

例)タスク生成
cre_taskからtk_cre_taskへ

【移行のポイント】
(1) タスク ID 格納処理の追加
(2) システムコール名称の変更
(3) タスク属性の追加
(4) スタック指定用メンバ変数名の変更

リスト 2-7 μITRON4.0 でのタスクの生成 (cre_tsk)

```
    T_CTSK   pk_ctsk;
    ER       ercd;
static INT   task_stack[0x400/sizeof(INT)];

        :

    pk_ctsk.exinf    = (VP_INT)0;
    pk_ctsk.tskatr   = TA_HLNG;
    pk_ctsk.task     = task;
    pk_ctsk.itskpri  = 1;
    pk_ctsk.stksz    = sizeof(task_stack);
    pk_ctsk.stk      = task_stack;

    ercd = cre_tsk( 10, &pk_ctsk );   /* タスク ID = 10 のタスク */

        :
```

```
    T_CTSK   pk_ctsk;
    ID       task_id;
static INT   task_stack[0x400/sizeof(INT)];

        :

    pk_ctsk.exinf    = (VP)0;
    pk_ctsk.tskatr   = TA_HLNG | TA_RNG0 | TA_USERBUF;
    pk_ctsk.task     = task;
    pk_ctsk.itskpri  = 1;
    pk_ctsk.stksz    = sizeof(task_stack);
    pk_ctsk.bufptr   = task_stack;

    task_id = tk_cre_tsk( &pk_ctsk );/* タスク ID は自動割当   */

        :
```

μITRONからμT-Kernelへの移行（その2）

■ラッパー関数などを利用して移行

◆システムコール名、固定IDと動的IDの変換などを自動的に変換

（例 1） メッセージバッファ生成

T-Kernel のメッセージバッファ ID の動的な値をラッパー内部で管理する配列 mbfid[] に保持するようにします。

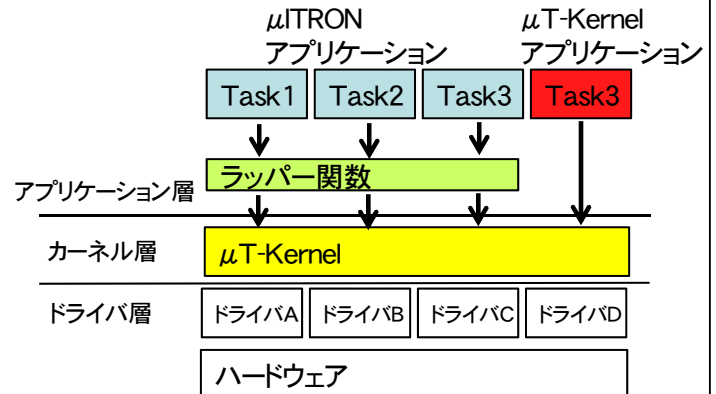
リスト 3-1 ITRON 用ソースプログラム（静的 API）

```
CRE_MBF(MBF_C, {TA_TFIFO, 10, 1000, NULL});
```

リスト 3-2 ラッパー内部の処理例

```
#define MBF_C  3  /* 固定 ID */

  T_CMBF cmbf = {NULL, TA_TFIFO, 1000, 10};
  mbfid[MBF_C - 1] = tk_cre_mbf(&cmbf);
```

†実際にはパラメータチェックやエラー処理等も必要です。

μT-Kernel対応状況

■2006年11月～一般にソースコード公開

■T-EngineフォーラムのHPからリファレンスコードを公開。パッチ含み

◆ARM7(μTeaboard) ― リファレンスコード

◆H8S/2212(UNL Active Tag)

◆FR60(CQ-FRK-FR60)

◆M16C(M16C/62P)

◆M32C(M32C/87)

◆V850-MA3(μT-Engine)

（H8S/2212 ～ V850-MA3：パッチで対応）

◆V850(TK-850/JG2)

■ユーシーテクノロジが各種最新MCUに対応した商用版 UCT μT-Kernelを2011年7月にリリース

UCT μT-Kernelを世界に向けて販売

■米国大手半導体・電子部品商社Avnetのオンラインストアを通じて世界に向けて販売

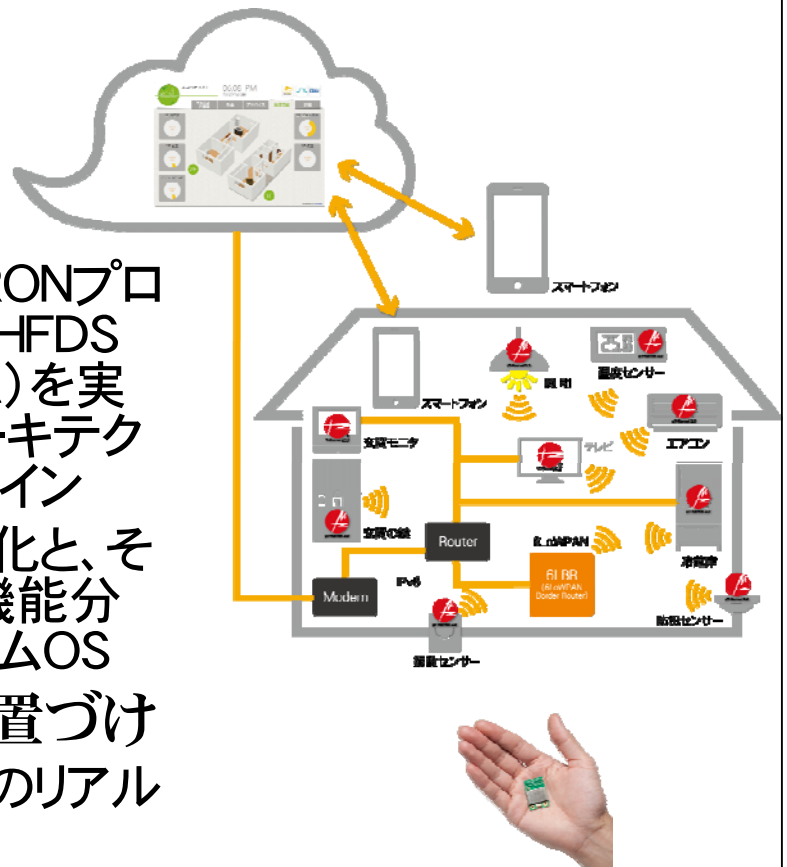
◆AvnetとARMが運営するオンラインストア「Embedded Software Store」からUCT μT-Kernelをダウンロード販売

● http://embeddedsoftwarestore.com/

μT-Kernel2.0とは？

μT-Kernelの“2.0”化

■T2全体コンセプト
- ◆1984年に開始したTRONプロジェクトの目標であるHFDS（超機能分散システム）を実現するための全体アーキテクチャの要素としてデザイン
- ⇒ネットワーク機能の強化と、それに基づく徹底した機能分散のためのリアルタイムOS

■μT-Kernel 2.0の位置づけ
- ◆M2M, IoTノードのためのリアルタイムOS

基本方針と設計コンセプト

■μT-Kernelの最新版
- ◆2013年12月11日 T-Engineフォーラムが全世界にむけて仕様書を公開
  - http://www.t-engine.org/ja/2013/download20131210.html
- ◆T2シリーズOS (T-Kernel 2.0, μT-Kernel 2.0, MP T-Kernel 2.0, …) 間の互換性を重視し、ソフトウェアの流通性を拡大
- ◆異なるMCUに実装されたμT-Kernel 2.0の差異を吸収し、共通に動作するソフトウェアを実現

■ T2μT1からの大幅な標準化範囲の拡大
■ サービスプロファイルの導入
■ 最適化・チューニングのための仕様更新
■ 割込み管理機能の整理と見直し

サービスプロファイルを導入

- μT-Kernel 2.0の実装仕様に関する情報を機械処理可能な形式で記述
  - C言語のマクロ定義によりヘッダファイルとして提供

```
/* 例: キャッシュ制御命令のサポート */
#define TK_SUPPORT_CACHE   TRUE
```

- サービスプロファイルを用いたコード記述を行うことで、異なるMCUに実装されたμT2間の差異が吸収され、ミドルウェアやアプリケーションコードの共通化が可能

サービスプロファイル一覧

プロファイル	内容
TK_STDTK	標準T-Kernel(必ずFALSE)
TK_MICROTK	μT-Kernel(必ずTRUE)
TK_ALLOW_MISALIGN	メモリのミスアラインアクセスが可能
TK_BIGENDIAN	ビッグエンディアン(定義必須)
TK_VIRTUAL_ADDRESS	仮想アドレスを利用(物理メモリ!=論理メモリ)
TK_VIRTUAL_MEMORY	仮想記憶が存在する(非常駐メモリが存在)
TK_TRAP_SVC	システムコールエントリにCPUのトラップ命令を利用
TK_HAS_DOUBLEWORD	64ビットデータ型(D, UD, VD)のサポート
TK_SUPPORT_SERCD	サブエラーコードのサポート
TK_HAS_SYSSTACK	タスクが独立したシステムスタックを持つ
TK_SUPPORT_FPU	FPU機能のサポート
TK_SUPPORT_COP0	番号0のコプロセッサ利用機能のサポート
TK_SUPPORT_COP1	番号1のコプロセッサ利用機能のサポート
TK_SUPPORT_COP2	番号2のコプロセッサ利用機能のサポート
TK_SUPPORT_COP3	番号3のコプロセッサ利用機能のサポート
TK_SUPPORT_RESOURCE	リソースグループのサポート
TK_SUPPORT_USERBUF	ユーザバッファ指定(TA_USERBUF)のサポート
TK_SUPPORT_AUTOBUF	自動バッファ割当て(TA_USERBUF指定なし)のサポート
TK_SUPPORT_SLICETIME	タスクスライスタイム設定(tk_chg_slt)のサポート
TK_SUPPORT_TASKINF	タスク統計情報取得機能(tk_inf_tsk)のサポート
TK_SUPPORT_USEC	マイクロ秒のサポート
TK_SUPPORT_TASKSPACE	タスク固有空間のサポート
TK_SUPPORT_TASKEVENT	タスクイベント機能のサポート
TK_SUPPORT_DISWAI	待ち禁止のサポート
TK_SUPPORT_REGOPS	レジスタの取得・設定機能のサポート
TK_SUPPORT_ASM	アセンブリによる処理ルーチンのサポート
TK_SUPPORT_DSNAME	DSオブジェクト名称のサポート
TK_SUPPORT_TASKEXCEPTION	タスク例外処理機能のサポート
TK_SUPPORT_LOWPOWER	省電力管理機能のサポート
TK_SUPPORT_SSYEVENT	サブシステムのイベント処理のサポート
TK_SUPPORT_LARGEDEV	大容量デバイス(64ビット)のサポート
TK_SUPPORT_INTCTRL	割込みコントローラ制御関連機能のサポート
TK_HAS_ENAINTLEVEL	割込みの優先度指定あり
TK_SUPPORT_CPUINTLEVEL	CPU内割込みマスクレベル取得・設定機能のサポート
TK_SUPPORT_CTRLINTLEVEL	割込みコントローラ内割込みマスクレベル設定・取得機能のサポート
TK_SUPPORT_INTMODE	割込みモード設定機能のサポート
TK_SUPPORT_SYSCONF	システム構成情報取得機能のサポート
TK_SUPPORT_IOPORT	I/Oポートアクセス機能のサポート
TK_SUPPORT_MICROWAIT	微小待ち機能のサポート
TK_SUPPORT_CACHECTRL	キャッシュ制御機能のサポート
TK_SUPPORT_WBCACHE	ライトバックキャッシュのサポート
TK_SUPPORT_WTCACHE	ライトスルーキャッシュのサポート
TK_SUPPORT_SYSMEMBLK	システムメモリ割当て機能のサポート
TK_SUPPORT_MEMLIB	メモリ割当てライブラリのサポート
TK_SUPPORT_ADDRSPACE	アドレス空間管理機能のサポート
TK_SUPPORT_PTIMER	物理タイマ機能のサポート

μT-Kernelから標準化範囲を拡大

■標準化範囲を拡大することで、T2シリーズOS間のミドルウェアやアプリケーションコードの共有を可能にする。

◆標準化された仕様

- 物理タイマ管理機能
- 微少待ち
- 高速ロック・マルチロック
- その他

最適化・チューニングのための仕様更新

■ CONSTの導入（T-Kernel 2.0仕様に準拠）

◆ 参照専用の変数であることをAPI仕様で明示

```
ID tskid = tk_cre_tsk( CONST T_CTSK *pk_ctsk );
```

◆ 読込み専用変数のROM領域への配置が可能になる

◆ ROM・RAMともに消費を削減することが可能になる

■ タスク優先度最大値に関する仕様を緩和

◆ 140（μT1）→ 16以上の値（μT2）

◆ タスク数が少ないシステムでは、スケジューラの最高優先度探索処理の効率化が可能

◆ RAM使用量を削減可能

■ 割込み関連の仕様を整理

◆「割込み番号」の概念を導入し、「割込みハンドラ番号」と「割り込みベクタ番号」を統一

■ その他、細かい仕様の見直し

◆ μT-Kernel 2.0仕様書のページ数

- 237ページ(μT1) → 639ページ

UCT μT-Kernel2.0

UCT μT-Kernel2.0の特長

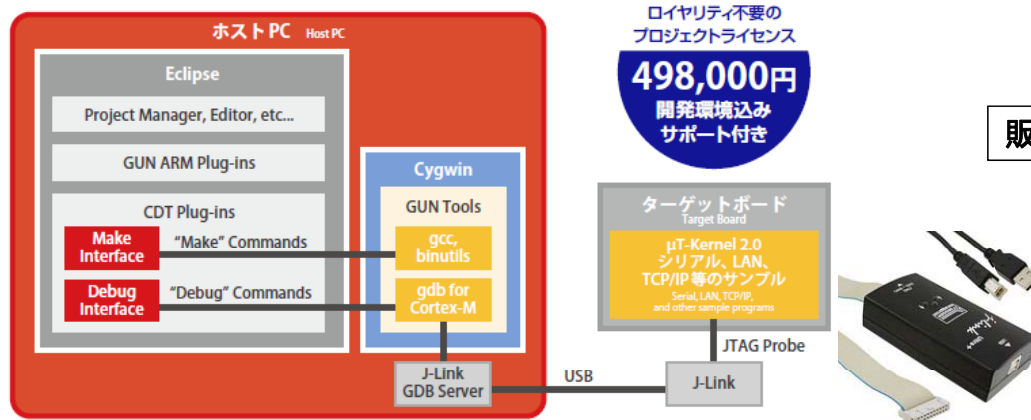
- 各社の最新マイコンに最適化したμT-Kernel 2.0とTCP/IPや各種サンプルソースコードを提供
  - μT-Kernel 1.01.02をベースに、μT-Kernel 2.0仕様に対応するように実装を拡張
  - 最小構成は、ROM 7KBとRAM 1KBで動作
  - ROM 36KB ＋ RAM 14KBで、簡易HTTPサーバが動作可能
    - μT-Kernel2.0、LANドライバ、簡易HTTPサーバDHCPクライアントを含む
- 各マイコンや開発環境プロジェクトファイルが付属。煩わしい設定作業が不要
    - MDK-ARM
    - EWARM
    - GCC/Eclips

UCT μT-Kernel 2.0 GCC開発キット

■ UCT μT-Kernel 2.0、Eclipse統合開発環境、GNU Cコンパイラ、JTAG-ICE(J-Link)など開発環境一式を提供

- ◆ TCP/IPプロトコルスタックや各種ドライバを含むサンプルコードが付属
  - LANドライバ、シリアルドライバ
  - ネットワーク、簡易シェル
  - 各種テンプレートサンプル
- ◆ 3ヵ月間のサポート付き
  - 6ヶ月単位で延長可能（98,000円（消費税別））
- ◆ ロイヤリティ不要で量産可能なライセンス（マイコン型名指定）
- ◆ 追加の開発環境（Eclipse+GNU Cコンパイラ+JTAG ICE）も198,000円（消費税別）で提供

eclipse

USB

ロイヤリティ不要の
プロジェクトライセンス
498,000円
開発環境込み
サポート付き

販売：イーソル株式会社

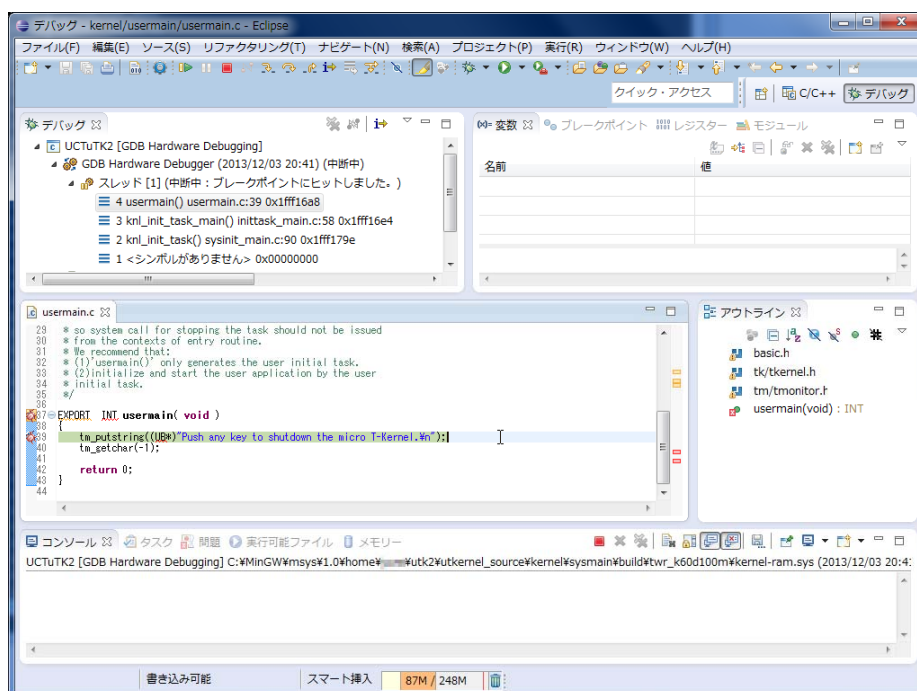
ターゲットボード Target Board
μT-Kernel 2.0
シリアル、LAN、
TCP/IP等のサンプル
Serial, LAN, TCP/IP, and other sample programs

JTAG Probe

J-Link

UC Technology

UCT μT-Kernel 2.0 GCC開発キット

■ 統合開発環境Eclipseで、ソースコードの開発からビルド、デバッグまでの一連の作業を行うことができます。

◆ Eclipse	4.3.1
◆ gcc	4.7.3
◆ gdb	7.6.1
◆ J-Link	V4.78c

UC Technology

J-TAG ICE (J-Link)

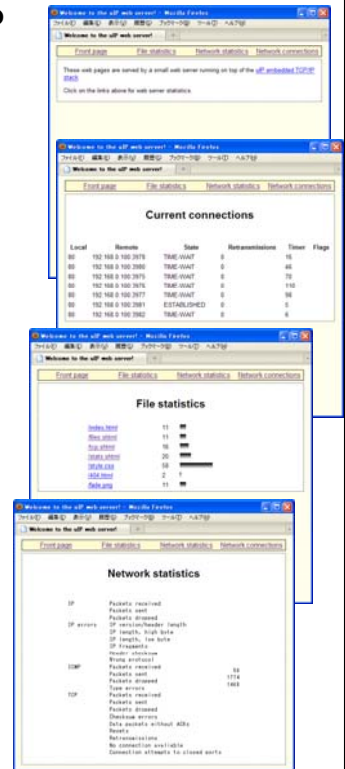
- **UCT μT-Kernel 2.0 GCC開発キットには、独Segger社製のJTAGエミュレータ(J-Link)を同梱**
  - ハードウェア構成
    - デバッガ本体 (J-Link)
    - 20ピンターゲットリボンケーブル
      - 他のケーブルが必要な場合は、お問い合わせください。
    - USBケーブル (A-B)
  - 詳細はこちら → http://www.segger.com/jlink_base.html
- **特徴**
  - OEMを含め、世界で最も利用されているJTAGエミュレータ
  - 多くのCPUコアに対応
  - 幅広い電源電圧に対応(1.2V～3.3V, 5Vに対応)
  - USB給電(専用の電源ケーブルが不要)、JTAGからボードへの給電も可能
  - RAMやフラッシュメモリーへの高速ダウンロード
    - ダウンロードスピード: 最大 1MB/sec
  - マイコンのフラッシュメモリーの中に無数の分断点を設定可能
    - Cortex-Mの外部フラッシュメモリーへの分断点の設定は、J-Linkの無制限フラッシュ分断点技術によってのみ可能

UCT μT-Kernel 2.0 GCC開発キット内容

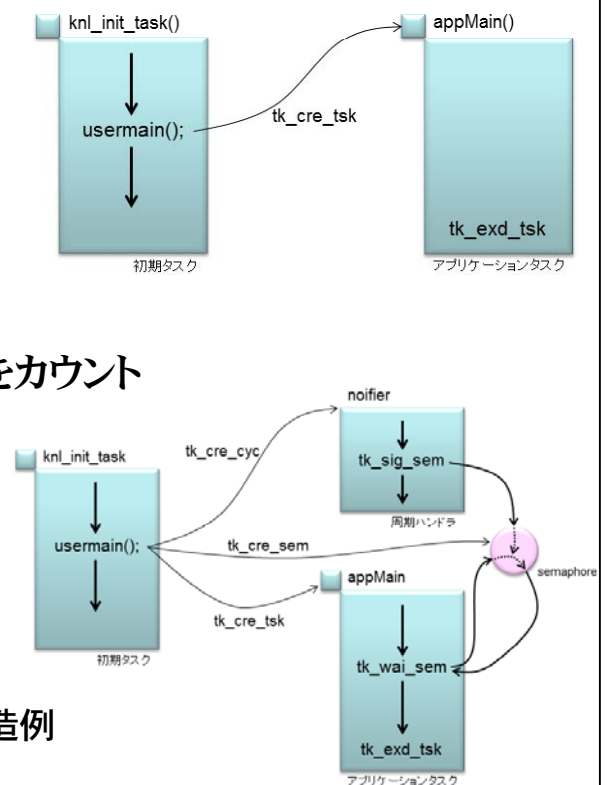
- **ソースコード**
  - μT-Kernel Ver.2.00.01
  - サンプルドライバ (RS-232Cドライバ、LANドライバ)
  - サンプルプログラム (ネットワーク、簡易シェル)
  - 各種テンプレートサンプル
- **Eclips / GCC開発環境**
  - Eclipse 4.3.1 (プロジェクト設定済)
  - MinGW
  - GNUツール、GDB
- **マニュアルなど**
  - UCT μT-Kernel 2.0取扱説明書
  - UCT μT-Kernel 2.0実装仕様書
  - UCT μT-Kernel 2.0ソースコード説明書 (utkernel.txt)
  - T-Engineデバイスドライバインタフェースライブラリ実装仕様書
  - RS-232Cドライバ実装仕様書
  - LANドライバ実装仕様書
  - uIP実装仕様書
  - UCTuTK2.0テンプレートサンプル説明書
  - μT-Kernel 仕様書 Ver. 2.00.00※
  - T-Engineデバイスドライバインタフェースライブラリ仕様Ver.1.00.00※
  - T-Engine標準デバイスドライバ仕様Ver.1.00.01※

    ※T-Engineフォーラムが公開している仕様書を同梱しています。

uIPネットワークサンプルソフト

- 組込み用の小型TCP/IPプロトコルスタックであるuIPを、μT-Kernel 2.0に移植しました。
  - uIPは、オープンなTCP/IPプロトコルスタックです。以下の機能がオープンソースとして公開されています。
    - 基本機能: TCP, UDP, IP, ARP, ICMP, SLIP, …
    - サンプルアプリケーション: HTTP client/server, SMTP client, telnet server, DNS hostname resolver, …
- ターゲットハードウェアに対応したLANドライバが付属していますので、すぐに動作させることが可能です。
  - T-Kernelのデバイスドライバ仕様を利用
- μT-Kenel 2.0のサンプルとして、以下を実装してあります。
  - 基本セット
ARP, IP, UDP, ICMP, TCP
  - サンプルアプリケーション
HTTP server, DNS hostname resolver , DHCP client

付属テンプレートサンプル例

- createTask: タスクの生成
  - 初期タスクから別のタスクを生成する方法
  - タスクの実装例
  - システムを終了する際の実装例
  - ソースコードをビルド対象に追加する際の改造例

- countAlarm: 周期ハンドラから起動回数をカウント
  - 初期タスクから別のタスクを生成する方法
  - タスクの実装例
  - セマフォの生成方法
  - セマフォの利用方法
  - 周期ハンドラの生成方法
  - エラー発生時の処理実装例
  - システムを終了する際の実装例
  - ソースコードをビルド対象に追加する際の改造例

UCT μT-Kernel 2.0開発キット

- UCT μT-Kernel 2.0 を市販の開発環境に対応
  - ◆ARM 純正開発（MDK-ARM）
  - ◆IAR システム社（EWARM）

ARM

- 各開発環境の環境設定ファイルが同梱で、ご使用の開発環境で直ぐに開発可能
- 各開発環境に対応したTCP/IPプロトコルスタックや各種ドライバを含むサンプルコード付属
  - ◆3ヵ月間のサポート付き
    - 6ヶ月単位で延長可能（98,000円（消費税別））
- ロイヤリティ不要で量産可能なライセンス（マイコン型名指定）
- 価格：398,000¥/ライセンス（消費税別）

UC Technology

μT-Kernel2.0対応ミドルウェアのご紹介

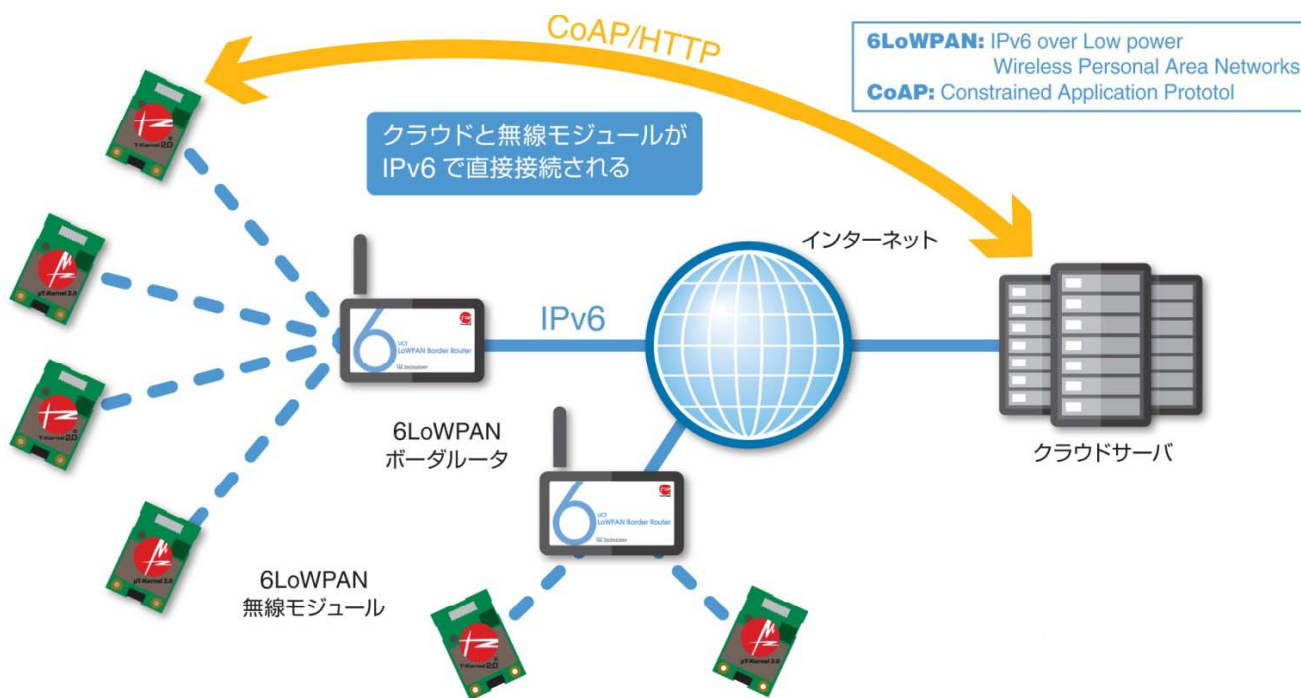
UCTμT-Kernel 2.0 対応ミドルウェア(1)

■ TCP/IPプロトコルスタック

- ◆組込みシステム向けに開発された軽量かつ高機能のTCP/IPプロトコルスタックで、多くの実績を持つlwIP v1.4.1 (lightweight TCP/IP)をUCT μT-Kernel 2.0に移植
- ◆BSDライクのソケットAPIのためアプリケーション開発が容易
- ◆修正BSDライセンスで自由に利用可能
- ◆必要な機能/性能に最適なコンフィグレーションが可能
  - ●20～40 KB ROM、10～20 KB RAM で動作
- ◆対応プロトコル
  - ●IPv4/IPv6, ICMP, UDP, TCP, ARP, IGMP, PPP, PPPoE, DHCP, AUTOIP, DNS, SNMP
- ◆提供形態
  - ●ソース提供
- ◆ライセンス
  - ●ロイヤリティー不要（マイコン型名指定）

UCTμT-Kernel 2.0 対応ミドルウェア(2)

■FATファイルシステム

- ◆FatFs R0.10 をUCT μT-Kernel 2.0に移植
- ◆小規模な組込みシステム向けに開発された超軽量のFATファイルシステムで独自 API
- ◆BSDスタイルの緩いライセンスで自由に利用可能
- ◆FAT12/16/32に対応
- ◆必要な機能/性能に最適なコンフィグレーションが可能
  - ●3～10 KB ROM、1～2 KB RAM で動作
- ◆LFN(Long File Name)にも対応可能
- ◆提供形態
  - ●ソース提供
- ◆ライセンス
  - ●ロイヤリティー不要（マイコン型名指定）

M2M/IoTクラウドベースシステム

CoAP/HTTP

6LoWPAN: IPv6 over Low power Wireless Personal Area Networks
CoAP: Constrained Application Prototol

クラウドと無線モジュールが
IPv6で直接接続される

インターネット

IPv6

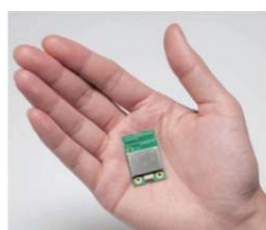
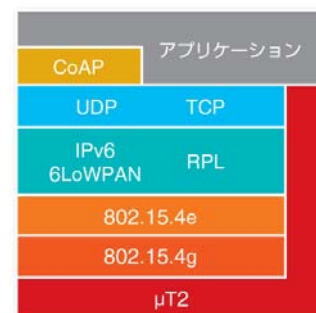
6LoWPAN
ボーダールータ

クラウドサーバ

6LoWPAN
無線モジュール

UC Technology

無線ミドルウェア 6LoWPANスタック

- 920MHzおよび2.4GHzの通信モジュール用の6LoWPANスタック
  - 小型無線通信モジュールにµT-Kernel 2.0、6LoWPAN、アプリケーションを含めて搭載
  - 典型的なアプリケーションを含めても、ワンチップマイコンに搭載可能なサイズを実現
    - ROM 20KB / RAM 12KB

6LoWPAN搭載920MHz無線通信モジュール

920MHz無線通信
モジュール用スタック

ユーシーテクノロジ株式会社

Ubiquitous Computing Technology Corporation

URL: http://www.uctec.com/

TEL: 03-5437-2323